Swarm Intell (2007) 1: 135–151
DOI 10.1007/s11721-007-0005-x

Ant colony optimization for real-world vehicle routing problems

From theory to applications

A.E. Rizzoli, · R. Montemanni · E. Lucibello · L.M. Gambardella

Received: 6 December 2006 / Accepted: 10 May 2007 / Published online: 13 September 2007
© Springer Science + Business Media, LLC 2007

Abstract Ant colony optimization (ACO) is a metaheuristic for combinatorial optimization problems. In this paper we report on its successful application to the vehicle routing problem (VRP). First, we introduce the VRP and some of its variants, such as the VRP with time windows, the time dependent VRP, the VRP with pickup and delivery, and the dynamic VRP. These variants have been formulated in order to bring the VRP closer to the kind of situations encountered in the real-world.

Then, we introduce the basic principles of ant colony optimization, and we briefly present its application to the solution of the VRP and of its variants.

Last, we discuss the applications of ACO to a number of real-world problems: a VRP with time windows for a major supermarket chain in Switzerland; a VRP with pickup and delivery for a leading distribution company in Italy; a time dependent VRP for freight distribution in the city of Padua, Italy, where the travel times depend on the time of the day; and an on-line VRP in the city of Lugano, Switzerland, where customers' orders arrive during the delivery process.

Keywords Ant colony optimization · Ant colony system · Vehicle routing problem · Dynamic VRP · Rich VRP · Real-world VRP

1 Introduction

The vehicle routing problem (VRP) concerns the transport of items between depots and customers by means of a fleet of vehicles. Examples of VRPs are: milk delivery, mail delivery, school bus routing, solid waste collection, heating oil distribution, parcel pick-up and delivery, dial-a-ride systems, and many others. Although finding the most cost efficient

A.E. Rizzoli, (✉) · R. Montemanni · L.M. Gambardella
Istituto Dalle Molle di Studi sull'Intelligenza Artificiale (IDSIA), Galleria 2, 6928 Manno, Switzerland
e-mail: andrea@idsia.ch

E. Lucibello
AntOptima, via Fusoni 4, 6900 Lugano, Switzerland

way to distribute goods across the logistic network is the main objective of supply-chain systems, only in the early '90s enterprise resource planning software vendors started to integrate tools to solve the VRP in supply chain management software (a review of software for supply chain management can be found in Aksoy and Derbez 2003).

The practical interest of the VRP has spawn a number of studies, which tackled the problem from many sides. Yet, the VRP is combinatorially complex and, therefore, as the size of the problem increases, it becomes harder and harder to obtain an exact solution for it in a reasonable amount of time. Thus, even the most advanced exact solution methods impose particular constraints on the problem instance, which are often violated when dealing with real-world vehicle routing problems, leaving practitioners unsatisfied with the performance and applicability of the algorithms.

Given the shortcomings of exact solution methods, researchers in the field of operations research (OR) started to develop *metaheuristics* (Blum and Roli 2003), heuristic methods that can be applied to a wide class of problems. One of the advantages metaheuristics have over traditional optimization algorithms is their ability to produce a good suboptimal solution in short time. The integration of optimization algorithms based on metaheuristics, such as tabu search (Glover and Laguna 1997), simulated annealing (Kirkpatrick et al. 1983), ant colony optimization (Dorigo et al. 1996, 1999), and iterated local search (Lourenço et al. 2003), with advanced logistic systems for supply chain management opens new perspectives for operations research applications in industry. In particular, for the solution of VRP and its variants, a number of metaheuristics have been successfully applied, such as: simulated annealing (Osman 1993), tabu search (Gendreau et al. 1994; Taillard et al. 1997), granular tabu search (Toth and Vigo 2003), genetic algorithms (Van Breedam 1996), guided local search (Kilby et al. 1999), variable neighborhood search (Bräysy 2003), greedy randomized adaptive search procedure (Resende and Ribeiro 2003), and ant colony optimization (Gambardella 1999; Reimann et al. 2003).

In this paper we focus on ant colony optimization (ACO) (Dorigo and Stützle 2004), a metaheuristic inspired by the foraging behavior of ant colonies. ACO has been used for the approximate solution of a number of traditional OR problems, among which the job shop scheduling problem, the quadratic assignment problem, the sequential ordering problem, the graph coloring problem, and the shortest common supersequence problem (Dorigo and Stützle 2004). More recently, ACO has been employed in a number of open shop scheduling problems (Blum 2005), in optimal product design (Albritton and McMullen 2007), and has also been used in some environmental problems, such as the design of a water distribution network (Zecchin et al. 2007) or the planning of wells for groundwater quality monitoring (Li and Chan Hilton 2007), thus proving its adaptability to very different domains of application.

The flexibility of the ACO metaheuristic allowed its application to many vehicle routing problems where heterogeneous vehicle fleets, limitations on customer accessibility, time windows, and the order imposed by pick-ups and deliveries considerably complicate the problem formulation. These kinds of problems have been labeled as *rich* vehicle routing problems (Hartl et al. 2006). Yet, real-world problems are even more complex; for instance, travel times may be uncertain and may depend on traffic conditions, and not all customers' orders may be perfectly known in time and dimension. These problem variants have been called *dynamic* VRP and they are currently attracting a lot of research efforts, because of their closeness to real-world traffic and distribution models (Zeimpekis et al. 2007). The objective of this paper is to describe how ant colony optimization can be successfully used to solve a number of VRP variants, both for some of the basic problem instances (the capacitated VRP, the VRP with time windows, the VRP with pickup and delivery) and for some

of the dynamic extensions (the time dependent vehicle routing problem and the on-line vehicle routing problem) where ACO has recently been applied and its ability to find efficient solutions in a short time has been proven useful also in this setting.

The paper is structured as follows: first we outline the VRP with its static and dynamic variants; then we introduce ACO, providing an overview of the main ACO algorithms; finally we discuss the application of ACO to the VRP. In this latter section, devoted to applications, we first introduce two large-scale industrial applications, showing how ACO can be successfully applied in the day-to-day operations of large real-world distribution processes, then we present the application of ACO to dynamic VRPs, showing its applicability in the context of urban freight distribution.

2 Vehicle routing problems

Finding optimal routes for a fleet of vehicles performing assigned tasks on a number of spatially distributed customers can be formulated as a combinatorial optimization problem: the vehicle routing problem. A solution of this problem is the best route serving all customers using a fleet of vehicles, respecting all operational constraints, such as vehicle capacity and the driver's maximum working time, and minimizing the total transportation cost.

The algorithm solving the problem requires the definition of an objective function that may include multiple objectives that are often conflicting. The most common objective is the minimization of transportation costs as a function of the traveled distance or of the travel time; the number of vehicles can be minimized expressing the costs associated with vehicles and drivers. Vehicle efficiency, expressed as the percentage of load capacity weighted by distance, can be taken into account. "Soft" constraints, which can be violated paying a penalty, can be included. For instance, if a customer is not served according to the agreed time schedule, a penalty might have to be paid. Road pricing schemes can also be considered, for example, attributing a higher cost to routes through city centers.

The elements that define and constrain each model of the VRP are: the *road network*, describing the connectivity among customers and depots; the *vehicles*, transporting goods between customers and depots on the road network; the *customers*, which place orders and receive goods.

The road network graph can be obtained from a detailed map of the distribution area on which the depots and the customers are geo-referenced. Standard algorithms can then be used to find the shortest routes between all couples of nodes in order to build the travel cost matrix. The matrix coefficients can represent the time required to travel from node i to node j, or the distance between the nodes, or any other metric that measures the travel cost. According to the adopted metric, different instances of the VRP may arise. For instance, if the travel time on edges depends on the time of the day, then we encounter the time dependent VRP (see Sect. 4.3.1 below).

The fleet of vehicles and their characteristics also impose constraints on the vehicle routing model. The fleet can be homogeneous, if all vehicles are equal in their characteristics, heterogeneous if this is not the case. Most real-world fleets are heterogeneous. Mechanical features (length, weight, width) and configuration (trailer, semi-trailer, van, etc.) constrain the ability of a vehicle to access road segments. For instance, a vehicle cannot travel on some arcs of a road network because of excessive weight or dimensions. On-board equipment, such as loading/unloading devices, may also impose constraints that depend on the type of customer to be served. Capacity constraints, stating the maximum load to be transported by a vehicle, are also important.

Each customer requests a given amount of goods, an order, which must be delivered or collected at the customer location. Time windows within which the customer must be served can be specified. These time windows can be single (only one continuous interval) or multiple (disjoint intervals). If time windows cannot be violated at any cost they are said to be "hard"; on the other hand, when a penalty is paid in case of violation, time windows are said to be "soft". Finally, the vehicle routing model can also include an estimation of the loading and unloading times at the customer—the so-called service time.

2.1 Basic problems of the vehicle routing class

Combining the various elements of the problem, we can define a whole family of different VRPs. Toth and Vigo (2001b) present a detailed overview of the various VRPs.

The capacitated vehicle routing problem (CVRP) is the basic version of the VRP. The name derives from the constraint of having vehicles with limited capacity. Customer demands are deterministic and known in advance. Deliveries cannot be split, that is, an order cannot be served using two or more vehicles. The vehicle fleet is homogeneous and there is only one depot. The objective is to minimize the total travel cost, usually expressed as the traveled distance required to serve all customers. The CVRP is NP-hard (Labbé et al. 1991) and the size of the problems which can be solved exactly in a reasonable time is up to 50 customers, using the branch-and-bound, branch-and-cut, and set-covering approaches (see Toth and Vigo 2001a).

When constraints on the delivery times are present, we have a vehicle routing problem with time windows (VRPTW): the capacity constraint still holds and each customer i is associated with a time window $[a_i, b_i]$ and with a service time s_i. VRPTW is also NP-hard, and even finding a feasible solution to the VRPTW is an NP-hard problem (Savelsbergh 1985). Good overviews on the VRPTW formulation and on exact, heuristic, and metaheuristic approaches to its solution can be found in Mester and Bräysy (2005), Li et al. (2005), and in Kytöjoki et al. (2007). Kallehauge et al. (2006) have solved VRPTW problems with 400 and 1000 customers by Lagrangian relaxation, but the problem formulation requires hard time windows.

In the VRP with pick-up and delivery (VRPPD) the transport items are not originally concentrated in the depots, but they are distributed over the nodes of the road network. A transportation request consists in transferring the demand from the pick-up point to the delivery point. These problems always include time windows for pick-up and/or delivery. A review of various approaches to the solution of the VRPPD is presented in (Desaulniers et al. 2000).

In all of these approaches the problem data is supposed not to change, neither during the planning phase (computation of the solution), nor during the management phase (implementation of the solution). More realistic situations might require the relaxation of this assumption.

2.2 Dynamic extensions of the VRP

The static formulations of the VRP, where customer demand is deterministic and travel times do not depend on the time of the day, have proven to be successful in modeling many practical problems. This is especially true for the VRPTW and VRPPD extensions. Yet, the availability of online information on the traffic conditions and the possibility of monitoring the vehicles' positions via the global positioning system, together with the online update of customer orders, can considerably change the problem settings. The availability of this

information has a price: the assumption of time-invariancy must be relaxed and data become time-dependent. Moreover, using data on current traffic conditions to estimate travel times requires the relaxation of the assumption of determinism, introducing uncertainty and adding another level of complexity to the problem.

In the literature these problems have been labeled with different terms: probabilistic, dynamic, and stochastic vehicle routing. These terms are often interchanged, but in practice we can assume that, in general, we are confronted with a dynamic vehicle routing problem, where problem data are often generated by a stochastic process. For instance, a stochastic process can be assumed to be responsible for the presence or absence of the customers, for the quantity of their orders, and for the travel and service times (Laporte and Louveaux 1998).

Stochastic customers and demands are typically formed when the planning horizon is longer than the horizon of the currently available data. These kinds of problems have been extensively studied (see, for instance, Gendreau et al. 1996); Bianchi et al. (2004) also studied how various metaheuristics can used in the solution of the VRP with stochastic demands.

One approach to solve the problem of unknown orders is to remove uncertainty by processing new orders as they come, in batches of variable size, according to what is called a *reactive* strategy. Potvin et al. (2006) list a number of reactive dynamic strategies for vehicle routing and scheduling problems; in the case of VRP this has been called the on-line variant (OLVRP). Among possible applications of the OLVRP we find *feeder systems*, which typically are local dial-a-ride systems aimed at feeding another, wider area, transportation system at a particular transfer location (Gendreau and Potvin 1998; Psaraftis 1988, 1995).

Another side of the problem is the presence of stochastic travel and service times, which are very frequent in urban environments. Especially with respect to travel times, the variability can be very high and considerably affect the solution. In the time dependent VRP variant (TDVRP) the variability can be reduced if one assumes that travel times are nearly constant within time periods in a day. This is quite true for peak and off-peak traffic conditions, which are observed in most cities. Ichoua et al. (2003) present a structured introduction to the problem and a model formulation.

3 The ACO metaheuristic and an overview of ACO algorithms

Ant colony optimization (Dorigo et al. 1999) is an optimization framework inspired by the observation, made by ethologists, that ants use *pheromone trails* to communicate information regarding shortest paths to food. A moving ant lays some pheromone (in varying quantities) on the ground, thus marking a path with a trail of this substance. An isolated ant moves mostly randomly and when it detects a previously laid pheromone trail it can decide, with high probability, to follow it, thus reinforcing the trail with its own pheromone. The collective behavior that results is a form of autocatalytic behavior where the more ants follow a trail, the more attractive for other ants it becomes. The process is thus characterized by a positive feedback loop, where the probability with which an ant chooses a path increases with the number of ants that previously chose the same path.

The above process inspired the ACO metaheuristic. The main elements are *artificial ants* (from now on simply ants), simple computational agents that individually and iteratively construct solutions for the problem, which has been modeled as a graph. Ants explore the graph visiting nodes connected by edges. A problem solution is an ordered sequence of nodes. The search process is executed in parallel over several constructive computational threads. A dynamic memory structure, inspired by the pheromone laying process, which

incorporates information on the effectiveness of previously obtained results, guides the construction process of each thread. Intermediate partial problem solutions are seen as *states*; at each iteration k of the algorithm each ant moves from state $x_k^{(i)}$ to $x_{k+1}^{(j)}$, expanding the partial solution from node i adding node j.

Ant system (AS) (Dorigo et al. 1996) was the first ACO algorithm to be proposed. It is organized in two main stages: construction of a solution, and update of the pheromone trail. In AS each ant builds a solution. An ant is in a given state and it computes a set of feasible expansions from it. The ant selects the move to expand the state taking into account the following two values:

1. the attractiveness η_{ij} of the move, as computed by some heuristic indicating the *a priori* desirability of that move;
2. the pheromone trail level τ_{ij} of the move, indicating how useful it has been in the past to make that particular move; it represents, therefore, an *a posteriori* indication of the desirability of that move.

Given the attractiveness and the pheromone trail level, an ant chooses to visit node j from node i according to the following probability:

$$p_{ij} = \begin{cases} \dfrac{[\tau_{ij}]^\alpha[\eta_{ij}]^\beta}{\sum_{h\in\Omega}[\tau_{ih}]^\alpha[\eta_{ih}]^\beta}, & \text{if } j \in \Omega, \\ 0, & \text{otherwise,} \end{cases} \tag{1}$$

where Ω is the set of nodes which can be visited starting from i. The parameters α and β weigh the influence of trails and attractiveness.

Once a solution is obtained, pheromone trails are updated. First, the pheromone is evaporated on all arcs, in order to progressively forget bad solutions; then all ants deposit pheromone on the arcs which are part of the solutions they have just computed.

Dorigo and Stützle (2004) distinguish between variants and extensions of AS. The variants differ from the original algorithm mainly in the pheromone update rule. Among variants we find: elitist ant system (Dorigo 1992); rank-based ant system (Bullnheimer et al. 1999); and $\mathcal{MAX}$-$\mathcal{MIN}$ ant system (Stützle and Hoos 2000), which uses bounding techniques to limit the possible values for pheromone on arcs. Extensions display more substantial changes in the algorithm structure. Among the extensions we find: approximate nondeterministic tree search (ANTS, Maniezzo and Carbonaro 2000) exploiting the use of lower bounds in the computation of a solution; D-ants (Reimann et al. 2002, 2004), which makes use of the savings algorithm; the hyper-cube framework for ACO (Blum and Dorigo 2004), which rescales pheromone values between 0 and 1; beam-ACO (Blum 2005), which hybridizes ACO with beam search; and, finally, ant colony system (ACS, Dorigo and Gambardella 1997) on which we focus in the remainder. In brief, ACS differs from AS for a revised rule used in the tour construction algorithm, and for the use of both local and global updates of the pheromone trails.

During tour construction ACS uses the *pseudo-random-proportional* transition rule. Let i be the last node visited in the partial solution under construction. If q is a random variable uniformly distributed over [0, 1], with probability q (*exploitation*) the next chosen node j is the one maximizing the product $\tau_{ij} \cdot \eta_{ij}$, while with probability $1 - q$ (*exploration*) the node j is chosen probabilistically via Monte Carlo sampling using (1). Notice that the pseudo-random-proportional rule provides a straightforward way to balance between exploration of new states and exploitation of a priori and accumulated knowledge. Parameter q, that is, the probability of carrying out exploitation rather than exploration, is usually set to 0.9.

ACS uses two different types of pheromone updates: local and global. Local update is performed every time an ant decides to use the edge from i to j in its solution; the pheromone τ_{ij} is then modified:

$$\tau_{ij} = (1-\rho)\cdot\tau_{ij} + \rho\cdot\tau_0, \tag{2}$$

where τ_0 is the initial pheromone value defined as $\tau_0 = 1/(n\cdot J_{nn})$, where n is the number of customers in the solution and J_{nn} the objective function produced by the execution of one ACS iteration without the pheromone component (this is equivalent to a probabilistic nearest neighbor heuristic).

Global update is carried out at the end of each iteration, when each ant has constructed a complete solution. The only edges modified are those belonging to the best solution generated so far, ψ^*. The objective function evaluates ψ^* to the value J^*. The global update formula is:

$$\tau_{i^*j^*} = (1-\rho)\cdot\tau_{i^*j^*} + \rho\cdot\frac{1}{J^*}, \quad \forall\big(i^*, j^*\big)\in\psi^*. \tag{3}$$

ACS has been shown to be very efficient in solving routing problems. In the next sections we show how it has been applied to the solution of some problem instances of the VRP class, ranging from the static case (VRP with time windows, and VRP with time windows and pick-up and delivery constraints) to the dynamic case (time dependent VRP and on-line VRP).

4 Solving the VRP with ACO

Sales and distribution processes require the ability to forecast customer demand and to optimally plan the distribution of the products to the consumers. These two strategic activities, forecast and optimization, must be tightly interconnected in order to improve the performance of the system as a whole (Gambardella et al. 1998).

In Fig. 1 the workflow of a distribution-centered company is sketched. The sales department generates new orders by contacting the customers (old and new) to check whether they need a new delivery. The effectiveness of this operation can be increased thanks to inventory management modules, which estimate the demand of every customer, indicating the best re-order time for each of them. New orders are then processed by the planning department, which, according to the quantities requested, the location of the customers, and the time windows for the delivery, decides how many vehicles to employ and computes the best routes for the delivery, in order to minimize the total travel time and space. This task is assisted by a vehicle routing algorithm, represented by the OPTIMIZE block. The vehicle tours are then assigned to the fleet, which is monitored by the fleet operational control station, which monitors the evolution of deliveries in real time. This process is assisted by the SIMULATE/MONITOR/RE-PLAN module, which allows re-planning online in face of new urgent orders, which were not available during the previous off-line planning phase. Finally, after vehicles have returned to the depot, delivery data are off-loaded and transferred back to the company database.

In the next sub-sections, we describe a number of real-world applications, where ACO has been used for the implementation of the vehicle routing algorithms that are executed in the OPTIMIZE block of Fig. 1.

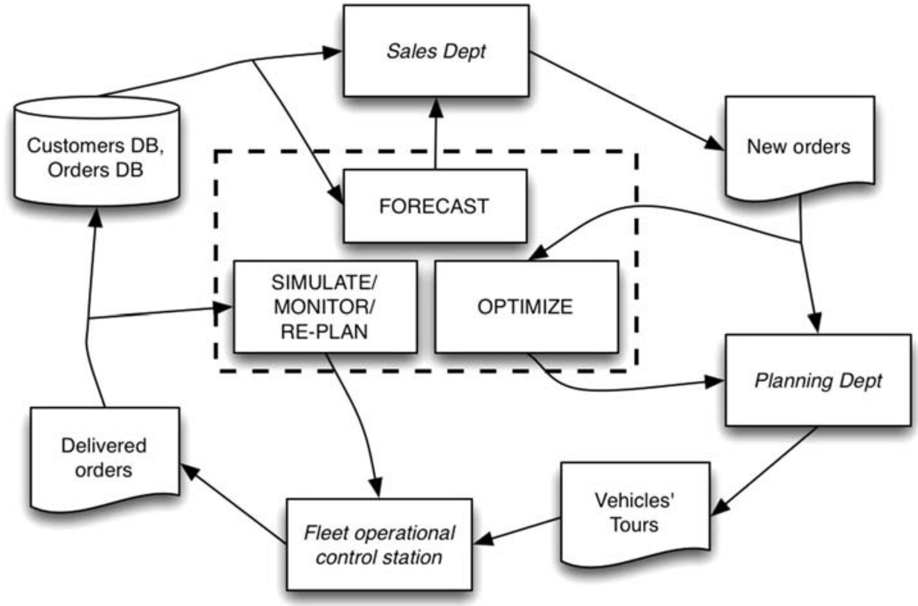

Fig. 1 The forecast–optimize–simulate loop: the role of optimization in the efficient management of a distribution process

4.1 A VRPTW application for the secondary level in the supply chain

In this application the client is one of the major supermarket chains in Switzerland. The problem is to distribute palletized goods to more than 600 stores, all over Switzerland. The stores order daily quantities of goods to replenish their local stocks. They want the goods to be delivered within time windows, in order to plan in advance the daily availability of their personnel, allocating a fraction of their time to inventory management tasks.

There are three types of vehicles: trucks (capacity: 17 pallets), trucks with trailers (35 pallets), and tractor units with semi-trailers (33 pallets). Whether a vehicle can access a store or not depends on the store location. In some cases the truck with trailer can leave the trailer at a previous store and then continue to other less accessible locations. The number of vehicles is assumed to be infinite, since transport services can be purchased on the market according to the needs.

The road network graph has been computed using digital road maps. The distances in kilometers between couples of stores have been rescaled using a speed model, which depends on the distance: longer distances allow a higher average speed. For instance, if the distance is less than 5 km, the average speed is 20 km/h; if the distance is more than 90 km, the speed is 60 km/h; in between there is a range of other speed values. The data have been collected over many years and they have been validated by the drivers' experience. The time to set-up a vehicle for unloading and the time required to hook/unhook a trailer are constant. The service time is variable and depends on the number of pallets to unload.

All the routes must be performed in one day, and the client imposes an extra constraint stating that a vehicle must perform its latest delivery as far as possible from the inventory, since it could be used to perform extra services on its way back. These extra services were not included in the planning by explicit request of the client.

4.1.1 The algorithm

This problem was modeled as a VRPTW, and solved by an implementation of the MACS-VRPTW algorithm (Gambardella 1999), named ANTROUTE. MACS-VRPTW is the most efficient ACO algorithm for the VRPTW and one of the most efficient metaheuristics overall for this problem. ANTROUTE adds to MACS-VRPTW the ability to handle the choice of the vehicle type: at the start of each tour the ant chooses a vehicle. A waiting cost was also introduced in order to prevent vehicles arriving too early at the stores.

The central idea of the MACS-VRPTW algorithm is to use two ant colonies (MACS stands for multi ant colony system). One colony, named ACS-VEI, minimizes the vehicles while the other one, named ACS-TIME, minimizes time. The two colonies are completely independent, since each one has its own pheromone trail, but they collaborate by sharing the variable ψ^*, which describes the best solution found so far. Each colony is composed of a number of ants. Every ant in the colony tries to build a feasible solution to the problem.

The algorithm works as follows: first compute a feasible initial solution ψ^*, with a number of vehicles v, using a nearest neighbor heuristic. ACS-VEI is then started: it tries to find a feasible solution ψ^* using $v-1$ vehicles. ACS-TIME is also started: given v vehicles, it tries to minimize the total time required to serve all the customers. When ACS-VEI finds a feasible solution with $v-1$ vehicles, ψ^* is updated and ACS-TIME is restarted with $v=v-1$. ACS-TIME serves the purpose of refining the solutions obtained by ACS-VEI, which has no feeling for the travel time, since its objective function is independent of it.

The constructive step of both colonies is based on a procedure that, starting from the current node i, computes the set of all feasible nodes. These are the nodes j still to be visited and such that the time of arrival at node j and the load are compatible with the time window and the delivery of the quantity q_j of goods requested by node j. The probabilistic choice is made according to the pseudo-random-proportional rule described in Sect. 3. In order to account for the effect of time windows, distance becomes a function of the time window: a node can appear "closer" if the end of its time windows is near. The end of the construction of a solution by all ants in the colony marks the end of one algorithm iteration. The pheromone trails are updated both *locally* and *globally* according to the ACS algorithm.

Note that in the ACS-VEI colony the ants usually construct infeasible solutions, that is, not all customers can be visited under the constraints given.

Given that the algorithm uses two colonies, at the end of each cycle, pheromone trails are globally updated for two different solutions: $\psi^{\text{ACS-VEI}}$, the infeasible solution with the highest number of visited customers, and ψ^*, the feasible solution with the lowest number of vehicles and the shortest travel time. Thus, pheromone is updated also on arcs that are not included in a feasible solution, which could still become feasible in the next iteration.

ACS-TIME uses a local search procedure to improve the quality of the feasible solutions, which is similar to the CROSS procedure (Taillard et al. 1997). Given that the CROSS procedure requires a feasible solution to operate on, such a local search can not be applied during ACS-VEI, where unfeasible solutions can still be returned at the end of a cycle. Both ACS-TIME and ACS-VEI try to repair infeasible solutions by inserting unvisited customers.

4.1.2 Results

The first tours computed by ANTROUTE were not accepted as feasible by the human tour planners, even if the performance was considerably higher than theirs and no explicit constraints were violated. Thus, a further modeling iteration was required, to let "hidden" constraints emerge. The human planners were actually using a regional planning strategy, that

Table 1 Comparison of the computer-generated vs. man-made tours in the VRPTW application

	Human planner	AR-RegTW	AR-Free
Total number of tours	2056	1807	1614
Total km	147271	143983	126258
Average truck loading	76.91%	87.35%	97.81%

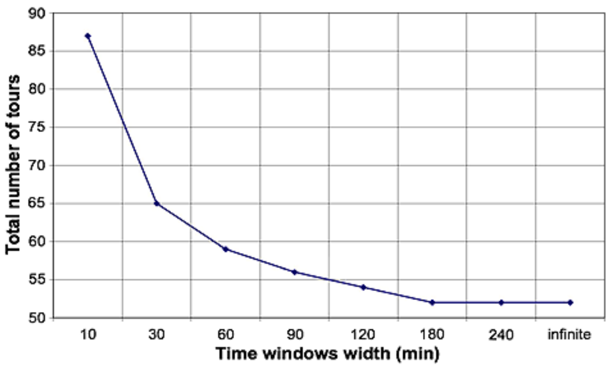

Fig. 2 The relationship between the time window width and the number of vehicle routes

led to petal shaped tours. This preference was included in the reformulation of the problem, but at the same time we tried to loosen the constraint a bit. We attributed stores to distribution regions, but at the same time we allowed stores near the border of the distribution region to also belong to the neighboring region. This allowed the generation of tours which were slightly worse than the unconstrained solution, but, nevertheless, better than the solutions found by the human planners. In Table 1 we present the results obtained by ANTROUTE compared with those of the human planners. ANTROUTE was run under two configurations: AR-RegTW, with regional planning and 1-hour time windows; AR-Free, where the regional and the time windows constraints were relaxed. The problem was to distribute 52000 pallets to 6800 customers over a period of 20 days. Every day ANTROUTE was run on the available set of orders and it took about 5 minutes to find a solution. At the same time, the planners were at work and it took them at least 3 hours to find a solution. At the end of the testing period, the performances of the algorithm and of the planners have been compared using the same objective function.

The advantage of an algorithm able to find the solution to an otherwise very hard problem in such a short time is the possibility of using it as a *strategic planning tool*. In Fig. 2 it is shown how running the algorithm with wider time-windows at the stores returns a smaller number of tours, which can be translated in a substantial reduction of transportation costs. The logistic manager can, therefore, use the optimization algorithm as a tool to investigate how to re-design the time-windows in the stores.

4.2 A VRPPD application for the primary level in the supply chain

In this application the client is a major logistics operator in Italy. The distribution process involves moving palletized goods from factories to inventory stores, before they are finally distributed to shops. A customer in this vehicle routing problem is either a pick-up or a delivery point. There is no central depot, and approximately 1000–1500 trucks per day are used. Routes can be performed within the same day, over two days, or over three days, since

the Italian peninsula is quite long and there's a strict constraint on the maximum number of hours per day that a driver can travel. All pick-ups of a tour must take place before deliveries. Orders cannot be split among tours. Time windows are associated with each store.

There is only one type of truck: tractor with semi trailer. The load is measured in pallets, in kilograms, and in cubic meters. There are capacity constraints on each one of these measurement units, and the first one that is exceeded triggers the violation of the constraint. The availability of vehicles is assumed to be infinite, since they are provided by flexible subcontractors. Sub-contractors are distributed all over Italy and, therefore, vehicles can start their routes from the first assigned customer, and no cost is incurred in traveling to the first customer in the route.

The road network graph has been elicited from digital road maps, computing the shortest path between each couple of stores. The travel times are computed according to the traveled distance, given the average speed that can be sustained on each road segment according to its type (highway, extraurban road, urban road). Loading and unloading times are assumed to be constant. This is a rough approximation imposed by the client, since he has been unable so far to provide better estimates. The client also imposed another constraint, related to the same problem, setting a maximum number of cities to visit per tour (usually less than six). Note that more than one customer can reside in a city. Moreover, the client requested that the distance between successive deliveries should be limited by a parameter.

4.2.1 The algorithm

The problem can be modeled as a VRP with pickup and delivery and time windows (VRP-PDTW). The objective function measures the average tour efficiency, $f = \sum_{i=1}^{N} e_i/N$, where e_i is the efficiency of tour i: the occupancy ratio of a vehicle over the traveled distance within the tour. It is computed according to the formula $e_i = \sum_{j=1}^{M_i} q_j l_j/(Q_i L_i)$, where M_i is the number of orders in the ith tour; q_j is the number of pallets in the jth order; l_j is the distance between source and destination points of the jth order; Q_i is the capacity of the vehicle serving the ith tour, and L_i is the total length of the ith tour.

The ANTROUTE algorithm has also been used in this context, but since there is a single objective—to maximize average efficiency—it has been adapted removing the ant colony minimizing the number of vehicles. The first step of an artificial ant is to select the starting city. Since this is a pickup and delivery problem, each source node must be paired with the corresponding destination node, and the search space is, therefore, harder to explore than in a delivery problem. The algorithm tries to simplify exploration using an approximation of the delivery phase, assuming that all deliveries will be performed in the reverse order with respect to pickups. Thus, a first stage local search exchanges nodes between tours, while preserving the order of deliveries; later, another local search procedure is applied, in which nodes are exchanged within the same tour.

4.2.2 Results

Table 2 summarizes the comparison between man-made and computer-generated tours over a testing period of two weeks. A noticeable improvement in the efficiency of computer-generated tours can be observed.

It is also interesting to remark how the algorithm performance is correlated with the difficulty of the problem, which is related to the number of orders to satisfy. In Fig. 3 we plot on the x-axis the efficiency of the man-made tours, and on the y-axis the efficiency improvement obtained using the computer-generated tours. When the problem is easy, because it

Table 2 Comparison of the computer-generated vs. man-made tours in the VRPPDTW application

	Human planner	ANTROUTE	Absolute difference	Relative difference
Total nr of tours	471.5	460.8	−10.7	−2.63%
Total km	175441	173623	−1818.2	−1.32%
Efficiency	84.08%	88.27%	+4.19%	–

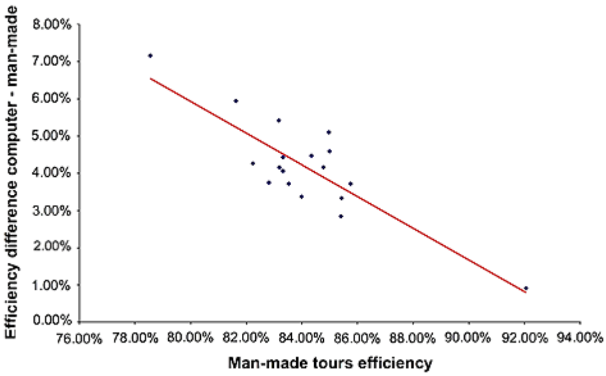

Fig. 3 Comparing man-made and computer-generated tours. Higher efficiency improvements are observed when the human planner performance is lower. The *dots* are experimental values, and the *solid line* is a regression on those values

involves a limited number of orders, and the human planner performs well, the computer is not able to provide a significative improvement, but when the planner starts to fail coping with the problem complexity, and the performance decreases, the gain in using the algorithm sensibly increases.

4.3 Time dependent VRPTW in the city of Padua

The city of Padua, Italy, set up a logistic platform to collect all incoming goods to be distributed to a number of shops in the city center. Such a platform aims at a better organization of the flow of goods into the city center, which is affected by traffic congestion problems and where loading and unloading space is scarcely available. Centralized planning of vehicle routes can sensibly reduce pollution and traffic problems due to commercial transport. For this purpose, Donati et al. (2007) have developed an algorithm solving the time dependent VRPTW for a logistic platform serving the city of Padua. In the study, the central depot was open from 8 am to 6 pm and traffic data on the Padua road network during that period was collected. Four time intervals, with similar traffic patterns and the relative travel speeds on network arcs, were identified. A set of 30 customers was considered.

4.3.1 The algorithm

The basic idea in the ACO algorithm of Donati et al. is to define a pheromone trail that is time dependent assuming that the travel times over the arcs of the graph depend on the time of the day. While the variation of travel times over time is continuous, it can be assumed that there are some distinctive time slices during one day when they are roughly constant. It is assumed that the duration of one working day can be partitioned in l time slices and, therefore, the pheromone trails can be described by $\tau_{ij}(l)$, with $l \in T_l$, where T_l is the set

of time slices into which the working time horizon is split. The objective is to minimize the total travel time.

The algorithm, based on MACS-VRPTW (see Sect. 4.1.1), then builds a solution making a probabilistic choice to select the next node j starting from i using the standard equation (1). The attractiveness η_{ij} of the next node is given by:

$$\eta_{ij}(t) = \frac{1}{f_{ij}(t) + w_j}, \tag{4}$$

where $f_{ij}(t)$ is the travel time from i to j evaluated at time $t \in T_l$ and w_j is the waiting time at node j.

Pheromone updating is carried out as described in Sect. 3, independently for the pheromone corresponding to each time slice.

4.3.2 Results

The authors compared the solution of the VRPTW using the time dependent variant with a solution of the same problem where the travel times on the road arcs were constant, depending only on the distance. In a series of nine tests, where customers were chosen randomly out of a set of real customers, it turned out that the time dependent variant performed 7% better than the standard VRPTW algorithm.

4.4 On-line VRP for fuel distribution

A leading Swiss fuel oil distribution company, which serves its customers from its main depot located near Lugano with a fleet of 10 vehicles, observed that during every Winter season there was always a subset of their customers that ran out of fuel and had to place urgent orders. These unexpected orders affects the planned delivery routes of the vehicles, and the vehicle routing problem becomes very "dynamic", since a noticeable percentage of orders must be fulfilled after the vehicles have already left the depot.

The objective of this study was to evaluate the impact of a reactive strategy for vehicle routing, starting from an analysis of the data collected in periods when urgent deliveries were in high demand. From the company data base, a sample of 50 customers was randomly selected, and travel times among them were calculated. In the company records, customers randomly appeared during the working day with random requests for a quantity of fuel to be delivered. A working day of 8 hours was considered, assuming a service time of 10 minutes for each customer. The cut-off time, after which the new orders received were postponed to the following working day, was set to 4 hours.

4.4.1 The algorithm

The problem description above matches the on-line VRP variant, where new orders can be assigned to vehicles which have already left the depot (e.g., parcel collection, feeder systems, fuel distribution, etc.).

To solve the on-line fuel oil distribution problem, Montemanni et al. (2005) have developed an ACO-inspired algorithm, ACS-DVRP, based on the decomposition of the on-line VRP into a sequence of static VRPs. There are three main elements in the algorithm architecture: the event manager, the ant colony algorithm, and the pheromone conservation strategy.

Table 3 Experimental results on the case study of Lugano

n_{ts}	200	100	50	25	10	5
t_{acs}	144	288	576	1152	2880	5760
t_{ls}	15	30	60	120	240	480
Travel time	12702	12422	10399	9744	10733	11201

The event manager receives new orders and keeps track of the already served orders and of the position and the residual capacity of each vehicle. This information is used to construct the sequence of static VRP-like instances. The working day is divided into time slices and for each of them a static VRP, which considers all the already received (but not yet executed) orders, is created. New orders received during a time slice are postponed until its end. At the end of each time slice, customers whose service time starts in the next time slice (according to the solution of the last static VRP) are assigned to the vehicles. They will not be taken into account in the following static VRPs.

The ACS algorithm employed is based on the one described in Sect. 4.1.1 for the VRPTW. The single ant colony is in charge of minimizing the total travel time.

Finally, the pheromone conservation strategy is based on the fact that, once a time slice is over and the relative static problem has been solved, the pheromone matrix contains information about good solutions. As each static problem is potentially very similar to the next one, this information is passed on to the next problem (Guntsch and Middendorf 2001): if a couple of customers appears in both the previous and the current time slice, the pheromone on the arcs connecting two nodes is brought forward as a fraction of its value in the previous problem.

4.4.2 Results

Algorithm ACS-DVRP was executed on a number of test problems, obtained varying the number n_{ts} of time slices into which the working day was divided. As the size of each problem in a time slice increases as the length of the time slice decreases, the time t_{acs} allocated to executing the ant colony system and the time t_{ls} dedicated to local search improving the solution were adjusted accordingly. In particular, the ratio between t_{acs} and t_{ls} was kept approximately equal to 10.

The first three rows of Table 3 define the settings of the experiments, that is, the values of parameters n_{ts}, t_{acs}, and t_{ls}. The fourth row shows the total travel time of the solutions found by the ACS-DVRP algorithm.

The results suggest that, for the case study analyzed, good values for n_{ts} are between 10 and 50. In particular, 25 seems to be the best choice. Large values of n_{ts} did not lead to satisfactory results because optimization was restarted too often, before a good local minimum could be reached. On the other hand, when n_{ts} was too small, the system was not able to take advantage of information on new incoming orders.

4.5 Conclusions

In this paper we have described how the ant colony optimization metaheuristic can be successfully used to solve a number of variants of the basic vehicle routing problem. We presented two industrial-scale applications of ACO for the solution of static VRP problems: a VRP with time windows and a VRP with pickup and delivery. We then focused our attention on two important dynamic variants of the VRP: the time dependent VRP, and the

on-line VRP. Both these problems are receiving increasing attention due to their relevance to real world problems, in particular, for distribution in urban environments.

In conclusion, after more than ten years of research, ACO has been shown to be one of the most successful metaheuristics for the VRP and its application to real-world problems demonstrates that it has now become a fundamental tool in applied operations research.

Acknowledgements The authors wish to acknowledge the contribution by F. Oliverio, of AntOptima, in the development of the ANTROUTE code. They also wish to thank the anonymous reviewers and the editors who greatly helped in improving the focus and organization of this paper.

References

Aksoy, Y., & Derbez, A. (2003). Software survey: supply chain management. *OR/MS Today*, *30*(3), 1–13.

Albritton, M. D., & McMullen, P. R. (2007). Optimal product design using a colony of virtual ants. *European Journal of Operational Research*, *176*(1), 498–520.

Bianchi, L., Birattari, M., Chiarandini, M., Manfrin, M., Mastrolilli, M., Paquete, L., Rossi-Doria, O., & Schiavinotto, T. (2004). Metaheuristics for the vehicle routing problem with stochastic demands. X. Yao, et al. (Eds.), *Lecture notes in computer science: Vol*. 3242. *Parallel problem solving from nature—PPSN VIII* (pp. 450–460). Berlin: Springer.

Blum, C. (2005). Beam-ACO—Hybridizing ant colony optimization with beam search: an application to open shop scheduling. *Computers and Operations Research*, *32*(6), 1565–1591.

Blum, C., & Dorigo, M. (2004). The hyper-cube framework for ant colony optimization. *IEEE Transactions on Systems, Man and Cybernetics—Part B: Cybernetics*, *34*(2), 1161–1172.

Blum, C., & Roli, A. (2003). Metaheuristics in combinatorial optimization: overview and conceptual comparison. *ACM Computing Surveys*, *35*(3), 268–308.

Bräysy, O. (2003). A reactive variable neighborhood search for the vehicle routing problem with time windows. *INFORMS Journal on Computing*, *15*(4), 347–368.

Bullnheimer, B., Hartl, R. F., & Strauss, C. (1999). A new rank-based version of the ant system: a computational study. *Central European Journal of Operations Research*, *7*(1), 25–38.

Desaulniers, G., Desrosiers, J., Erdmann, A., Solomon, M. M., & Soumis, F. (2000). VRP with pickup and delivery. In P. Toth & D. Vigo (Eds.), *The vehicle routing problem* (pp. 225–242). Philadelphia: SIAM.

Donati, A. V., Montemanni, R., Casagrande, N., Rizzoli, A. E., & Gambardella, L. M. (2007, in press). Time dependent vehicle routing problem with a multi ant colony system. *European Journal of Operational Research*.

Dorigo, M. (1992). *Optimization, learning and natural algorithms*. PhD thesis, Dipartimento di Elettronica, Politecnico di Milano, Italy.

Dorigo, M., & Gambardella, L. M. (1997). Ant colony system: A cooperative learning approach to the traveling salesman problem. *IEEE Transactions on Evolutionary Computation*, *1*(1), 53–66.

Dorigo, M., & Stützle, T. (2004). *Ant colony optimization*. Cambridge: MIT Press.

Dorigo, M., Maniezzo, V., & Colorni, A. (1996). Ant system: optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man and Cybernetics—Part B: Cybernetics*, *26*(1), 29–41.

Dorigo, M., Di Caro, G., & Gambardella, L. M. (1999). Ant algorithms for discrete optimization. *Artificial Life*, *5*, 137–172.

Gambardella, L. M., Rizzoli, A. E., & Zaffalon, M. (1998). Simulation and planning of an intermodal container terminal. *Simulation*, *71*(2), 107–116.

Gambardella, L. M., Taillard, É., & Agazzi, G. (1999). MACS-VRPTW: a multiple ant colony system for vehicle routing problems with time windows. In D. Corne, M. Dorigo & F. Glover (Eds.), *New ideas in optimization* (pp. 63–76). London: McGraw–Hill.

Gendreau, M., & Potvin, J.-Y. (1998). Dynamic vehicle routing and dispatching. In T. G. Crainic & G. Laporte (Eds.), *Fleet management and logistic* (pp. 115–226). Berlin: Springer.

Gendreau, M., Hertz, A., & Laporte, G. (1994). A tabu search heuristic for the vehicle routing problem. *Management Science*, *40*(10), 1276–1290.

Gendreau, M., Laporte, G., & Séguin, R. (1996). Stochastic vehicle routing. *European Journal of Operational Research*, *88*(1), 3–12.

Glover, F., & Laguna, M. (1997). *Tabu search*. Boston: Kluwer Academic.

Guntsch, M., & Middendorf, M. (2001). Pheromone modification strategies for ant algorithms applied to dynamic TSP. In E. J. W. Boers et al. (Eds.), *Lecture notes in computer science: Vol*. 2037. *Applications of evolutionary computing: EvoWorkshops 2001: EvoCOP, EvoFlight, EvoIASP, EvoLearn, and EvoSTIM* (pp. 213–222), Como, Italy, 18–20 April 2001. Berlin: Springer.

Hartl, R. F., Hasle, G., & Janssens, G. K. (2006). Special issue on rich vehicle routing problems: editorial. *Central European Journal of Operations Research*, *14*(2), 103–104.

Ichoua, S., Gendreau, M., & Potvin, J.-Y. (2003). Vehicle dispatching with time-dependent travel times. *European Journal of Operational Research*, *144*(2), 379–396.

Kallehauge, B., Larsen, J., & Madsen, O. B. G. (2006). Lagrangian duality applied to the vehicle routing problem with time windows. *Computers and Operations Research*, *33*(5), 1464–1487.

Kilby, P., Prosser, P., & Shaw, P. (1999). Guided local search for the vehicle routing problem. In S. Voss, S. Martello, I. H. Osman & C. Roucairol (Eds.), *Meta-heuristics: advances and trends in local search paradigms for optimization* (pp. 473–486). Boston: Kluwer Academic.

Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by simulated annealing. *Science*, *220*(4598), 671–680.

Kytöjoki, J., Nuortio, T., Bräysy, O., & Gendreau, M. (2007). An efficient variable neighborhood search heuristic for very large scale vehicle routing problems. *Computers and Operations Research*, *34*(9), 2743–2757.

Labbé, M., Laporte, G., & Mercure, H. (1991). Capacitated vehicle routing on trees. *Operations Research*, *39*(4), 616–622.

Laporte, G., & Louveaux, F. V. (1998). Solving stochastic routing problems with the integer L-shaped method. In T. G. Crainic & G. Laporte (Eds.), *Fleet management and logistics* (pp. 159–167). Boston: Kluwer Academic.

Li, Y., & Chan Hilton, A. B. (2007). Optimal groundwater monitoring design using an ant colony optimization paradigm. *Environmental Modelling and Software*, *22*(1), 110–116.

Li, F., Golden, B., & Wasil, E. (2005). Very large-scale vehicle routing: new test problems, algorithms, and results. *Computers and Operations Research*, *32*(5), 1165–1179.

Lourenço, H. R., Martin, O., & Stützle, T. (2003). Iterated local search. In F. Glover & G. Kochenberger (Eds.), *Handbook of metaheuristics* (pp. 321–353). Boston: Kluwer Academic.

Maniezzo, V., & Carbonaro, A. (2000). ANTS heuristic for the frequency assignment problem. *Future Generation Computer Systems*, *16*(8), 927–935.

Mester, D., & Bräysy, O. (2005). Active guided evolution strategies for the large scale vehicle routing problem with time windows. *Computers & Operations Research*, *32*(6), 1593–1614.

Montemanni, R., Gambardella, L. M., Rizzoli, A. E., & Donati, A. V. (2005). Ant colony system for a dynamic vehicle routing problem. *Journal of Combinatorial Optimization*, *10*, 327–343.

Osman, I. H. (1993). Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem. *Annals of Operations Research*, *41*, 421–451.

Potvin, J.-Y., Xu, Y., & Benyahia, I. (2006). Vehicle routing and scheduling with dynamic travel times. *Computers and Operations Research*, *33*(4), 1129–1137.

Psaraftis, H. (1995). Dynamic vehicle routing: status and prospects. *Annals of Operations Research*, *61*, 143–164.

Psaraftis, H. (1988). Dynamic vehicle routing problems. In B. L. Golden & A. A. Assad (Eds.), *Vehicle routing: methods and studies* (pp. 223–248). Amsterdam: North-Holland.

Reimann, M., Doerner, K., & Hartl, R. F. (2002). A savings based ant system for the vehicle routing problem. In W. B. Langdon et al. (Eds.), *Proceedings of the genetic and evolutionary computation conference (GECCO-2002)* (pp. 1317–1325). San Francisco: Kaufmann.

Reimann, M., Doerner, K., & Hartl, R. F. (2003). Analyzing a unified ant system for the VRP and some of its variants. In G. Raidl et al. (Eds.), *Lecture notes in computer science: Vol.* 2611. *Applications of evolutionary computing: EvoWorkshops 2003: EvoBIO, EvoCOP, EvoIASP, EvoMUSART, EvoROB, and EvoSTIM* (pp. 300–310), Essex, UK, 14–16 April 2003. Berlin: Springer.

Reimann, M., Doerner, K., & Hartl, R. F. (2004). D-ants: savings based ants divide and conquer the vehicle routing problem. *Computers and Operations Research*, *31*(4), 563–591.

Resende, M. G. C., & Ribeiro, C. C. (2003). Greedy randomized adaptive search procedures. In F. Glover & G. Kochenberger (Eds.), *Handbook of metaheuristics* (pp. 219–249). Boston: Kluwer Academic.

Savelsbergh, M. W. P. (1985). Local search in routing problems with time windows. *Annals of Operations Research*, *4*, 285–305.

Stützle, T., & Hoos, H. H. (2000). MAX-MIN ant-system. *Future Generation Computer Systems*, *16*(8), 889–914.

Taillard, È. D., Badeau, E. P., Gendreau, M., Guertin, F., & Potvin, J.-Y. (1997). A tabu search heuristic for the vehicle routing problem with soft time windows. *Transportation Science*, *31*(2), 170–186.

Toth, P., & Vigo, D. (2001a). Branch-and-bound algorithms for the capacitated VRP. In P. Toth & D. Vigo (Eds.), *The vehicle routing problem* (pp. 29–51). Philadelphia: SIAM.

Toth, P., & Vigo, D. (2001b). An overview of vehicle routing problems. In P. Toth & D. Vigo (Eds.), *The vehicle routing problem* (pp. 1–26). Philadelphia: SIAM.

Toth, P., & Vigo, D. (2003). The granular tabu search and its application to the vehicle routing problem. *INFORMS Journal on Computing*, *15*(4), 333–346.

Van Breedam, A. (1996). *An analysis of the effect of local improvement operators in genetic algorithms and simulated annealing for the vehicle routing problem*. RUCA Working Paper 96/14, University of Antwerp, Belgium, 1996.

Zecchin, A. C., Maier, H. R., Simpson, A. R., Leonard, M., & Nixon, J. B. (2007). Ant colony optimization applied to water distribution system design: comparative study of five algorithms. *Journal of Water Resources Planning and Management*, *133*(1), 87–92.

Zeimpekis, V., Tarantilis, C. D., Giaglis, G. M., & Minis, I. (2007). *Dynamic fleet management—concepts, systems, algorithms & case studies*. Berlin: Springer.